REMARKS

Claims 1-7, 9, 12, 63-65, 78-84, 86, 89-92, 108 and 109 were rejected as being unpatentable over Lavoie and Riedel. Reconsideration and withdrawal of these rejections are respectfully requested.

Independent claim 1 and its dependent claims

Independent claim 1 recites:

> **...wherein each of the at least two central servers, upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet, is configured to send a single outbound game payload to the gaming machine having sent the instance of the transaction packet, the outbound game payload enabling the gaming machine having sent the instance of the transaction packet to complete the game transaction and wherein the at least one gaming machine is configured such that a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when a second later arriving outbound payload is received by the at least one gaming machine.**

The Office Action acknowledges that Lavoie does not teach this subject matter and points to Riedel for a teaching/suggestion thereof.

However, a close reading of Riedel compels a different interpretation. In paragraph **[0019]**, Riedel et al. state that the "[s]ervers 130-150 may also synchronize responses, such that only a single response is provided for each request." (emphasis added). Riedel also proposes to group the servers into subsets and assign a synchronization server to determine "which server's response will be transmitted to the client"

An alternative arrangement is disclosed in Riedel at paragraph **[0020]** in which, contrary to the single responding server of paragraph **[0019]**, "Multiple identical response may then be transmitted to the client and the client may store information provided with the first received response and drop subsequent responses"

However, Riedel cautions us:

> **[0020] Alternatively, for operations that do not change the state of the system, but only return data to a client (e.g., READ and GETATTR for NFS), responses need not be synchronized. A request that requires returning data to the client is multicast to the group, and multiple servers in a subset may respond to the same request. Multiple identical responses may then be transmitted to the client, and the client may store information provided with the first received response and drop subsequent identical responses.**

Indeed, Riedel cautions the reader that **for operations that do not change the state of the system**, response need not be synchronized, multiple identical response may be transmitted to the client, who may then store the requested information in the first received response and drop the others. An example of operations that do not change the state of the system and are appropriate for multiple response is the READ and GETATTR commands. The READ command is, well, a "read" command and the GETATTR is a command that returns the attributes associated with a file, folder or directory. Therefore, as providing this information does not change the state of the system, multiple response by the servers to the clients are allowed.

In direct contrast, it is undeniable that the claimed outbound payloads sent from each of the at least two central servers do, in fact, "change the state of the system" (to use Riedel's own words), for the simple reason that, as claimed, "**a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when a second later arriving outbound payload is received by the at least one gaming machine**"

Therefore, because Riedel unequivocally teaches that his unsynchronized responses in which several servers respond to a request are only suitable for operations that do not change the state of the system, and because "completing a game transaction" very much does change the state of the system, the person of skill in the art, even in possession of Lavoie, would not look to Riedel for a teaching or suggestion on ways to modify the Lavoie system, because Riedel is categorical in his teaching that the multiple responding servers scheme should only be used for operations (such as a READ operation or a GETATTR command) that do **not** change the state of the system.

It is, therefore, respectfully submitted that the applied combination of Lavoie and Riedel would not have taught or suggested the embodiment of claim 1 to a person of ordinary skill in the art. To the contrary, the invention defined by claim 1, would have been wholly unsuggested to the skilled artisan, given the clear teachings of Riedel that limit such multiple responding servers scheme to situations where the state of the system is unchanged – which is clearly not the case in situations in which the receipt of an outbound payload completes a game transaction, which undisputedly changes the state of the system. Reconsideration and withdrawal of the 35 USC §103(a) rejections applied to claim 1 and its dependent claims are, therefore, respectfully requested.

Independent claim 79 and its dependent claims

Independent claim 79 recites:

> **at least one gaming machine coupled to the communication network, each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a separate instance of a single transaction packet from the at least one gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent to each of the at least two central servers including an identical inbound game payload,** ***wherein each of the two central servers are configured such that any transaction packet that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a***

> ***responding one of the at least two central servers to the synchronization engine of the non-responding central server. (Italics added for emphasis)***

Kindly note here that it is the synchronization engine of a responding one of the central servers that sends the transaction packet to the synchronization engine of a non-responding central server.

Several points are of note at this juncture, regarding claim 79:

1. Each of the central servers includes a synchronization engine;

2. The Synchronization engine of a responding central server sends the transaction packet to the synchronization engine of a non-responding server.

The Lavoie/Riedel combination does not teach or suggest any gaming system as defined in independent claim 79.

At the outset, let us examine paragraph **[0019]** of Riedel:

> **[0019] Servers 130-150 may also synchronize responses, such that only a single response is provided for each request. For example, servers 130-150 may be divided into two subsets, each subset associated with a request type (e.g., a data subset including data servers 140 and 150, and a meta data subset including meta data servers 130 and 135). A synchronization server may be designated for each server subset, such that synchronization may be performed by a minimal number of servers (i.e., the servers in a relevant subset). Synchronization may include reliably broadcasting the multicast message to each server in the subset in order to insure that each server has received the message. Also, synchronization may include transmitting a single response for each request. For example, two mirrors may respond to a read request, because each mirror stores the requested data. The synchronization server may determine, for example, at the instance of receipt of the request, which server's response will be transmitted to the client. The decision may be based on a predetermined arrangement or based on a least congested communication path to the client, as described above. Also, one server in the subset may respond to the request, and reliably broadcast a message to each server in the subset indicating that a response to that request has been transmitted. The servers receiving the message may then delete that request from their respective queues. (Underlining for emphasis)**

It is respectfully submitted that Riedel, whether considered singly or in combination with Lavoie, does not teach or suggest any gaming system in which each of the central servers includes a synchronization engine and the synchronization engine of a responding central server sends the transaction packet to the synchronization engine of a non-responding server.

Indeed, Riedel teaches that synchronization means several things:

1. To synchronize responses, such that only a single response is provided for each request;
2. To broadcast a multicast message from a synchronization server to each server in a subset of servers;
3. To determine which server's response will be transmitted to the client, and
4. To broadcast a message to the server(s) that did not transmit a response to the client that a response has, in fact, been transmitted.

However, claim 79 recites "*wherein each of the two central servers are configured such that any transaction packet that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server.*"

Kindly note the precise language: the transaction packet that is not acknowledged by a non-responding central server is just that: a transaction packet by which a gaming machine commits a transaction to the central servers, so function (1) of Riedel above would be of no help to a person of ordinary skill in this instance, since (1) refers to <u>responses</u> to requests). Riedel, moreover, uses a designated synchronization server for each subset of servers, whereas the claimed embodiment calls for <u>each</u> of the central servers to include a synchronization engine, so function (2) of Riedel also would not teach or suggest the claimed embodiment to a person of skill in the art. The claimed embodiment does not call for a determination of which central server will respond to the transaction packets sent by the gaming machines to the central servers, so function (3) above is would provide no teaching or suggestion helpful to the skilled artisan in developing the claimed embodiment, even if such skilled artisan were in full possession of both Lavoie and Riedel. Lastly, the claimed embodiment requires that the synchronization engine of a responding central server send a transaction packet (previously received from a gaming machine) to the synchronization engine of a non-responding central server. The claimed embodiment does <u>not</u> require a responding server to broadcast a message to the server(s) that did not transmit a response to the client that a response has, in fact, been transmitted, as defined by function (4) above.

Riedel, therefore, whether considered singly or in combination with Lavoie, therefore, does not teach or suggest the claimed embodiments of claim 79 and of its dependent claims.

Unless the Office points to a reference or a combination of references that teach or suggest two or more central servers being configured such that any transaction packet that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server, the outstanding 35 USC §103(a) is believed to be untenable and should be withdrawn. The same is, therefore, respectfully requested.

<u>Independent claim 108 and its dependent claims</u>

Claim 108 recites:

> **committing each game transaction to each of the at least two central servers by sending a separate instance of a single transaction packet from the at least one gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent to each of the at least two central servers including an identical inbound game payload;**
>
> **returning, by each of the at least two central servers, a single outbound game payload to the gaming machine upon receipt of the inbound**

> **game payload from the gaming machine having sent the instance of the transaction packet, and**
>
> **completing the game transaction, by the gaming machine, upon receipt of a first in time received outbound game payload from one of the at least two central server, irrespective of when a later in time outbound game payload is received from another one or ones of the at least two central servers.**

It is respectfully submitted that the comments above relative to independent claim 1 are equally applicable here. Therefore, the comments above relative to independent claim 1 are incorporate here by reference, as if repeated here in full.

<u>Independent claim 109 and its dependent claims</u>

Independent claim 109 recites:

> **109. (Previously Presented) A computer-implemented method of carrying out a game transaction, comprising the steps of:**
>
> **providing at least two central servers;**
>
> **providing a gaming machine configured to enable a game to be played;**
>
> **coupling the gaming machine and each of the at least two provided central servers to a communication network**
>
> **carrying out a game transaction for each game played;**
>
> **committing each game transaction to each of the at least two central servers by sending, by the gaming machine, a separate instance of a single transaction packet from the gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent from the gaming machine to each of the at least two central servers including an identical inbound game payload;**
>
> **completing the game transaction, by the gaming machine, upon receipt of a first in time received outbound game payload from one of the at least two central server, irrespective of when a later in time outbound game payload is received from another one or ones of the at least two central servers, and**
>
> ***recording, in the gaming machine, a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding one of the at least two central servers after a predetermined timeout, the synchronization log being configured to enable the gaming machine to subsequently send the unacknowledged transactions to the non-responding one of the at least two central servers. (Italics added for emphasis)***

Kindly note here that it is the gaming machines that record a synchronization log that enables the gaming machine to subsequently send any unacknowledged transactions to the non-responding central servers(s). The Lavoie/Riedel combination does not teach or suggest any gaming machines that record a synchronization log that enables the gaming machines to send any unacknowledged transactions to the non-responding central servers(s). Riedel does not teach or suggest that any of the clients disclosed therein have any synchronization duties whatsoever. In Riedel, the clients simply multi-cast their requests to the servers and the server take care of any needed synchronization, in the many ways that Riedel uses that term.

A person of ordinary skill, in full possession of both the Lavoie and Riedel references, would find no guidance, teaching or suggestion in the applied combination to devise a method in which the gaming machines recorded a log that enabled them to synchronize non-responding servers, as called for and required by claim 108. Therefore, it is respectfully submitted that the 35 USC §103(a) rejection should be reconsidered and withdrawn. The same is, therefore, respectfully requested.

New independent claim 111 includes the subject matter of independent claim 1 and that of allowable dependent claim 11.

Applicants' attorney believes that the present application is now in condition for allowance and passage to issue. If any unresolved issues remain, the Examiner is respectfully invited to contact the undersigned attorney of record at the telephone number indicated below, and whatever is required will be done at once.

Respectfully submitted,

Date: June 24, 2011 By:

Alan W. Young
Attorney for Applicants
Registration No. 37,970

YOUNG LAW FIRM, P.C.
4370 Alpine Rd., Ste. 106
Portola Valley, CA 94028
Tel.: (650) 851-7210
Fax: (650) 851-7232

C:\YLF\CLIENTS\CYBS\5872 (Univ Gaming Server)\5872 Amendment REsp to Non-Final OA of 04-11-2011.doc